

## 多语言和对话

### 多语言部分

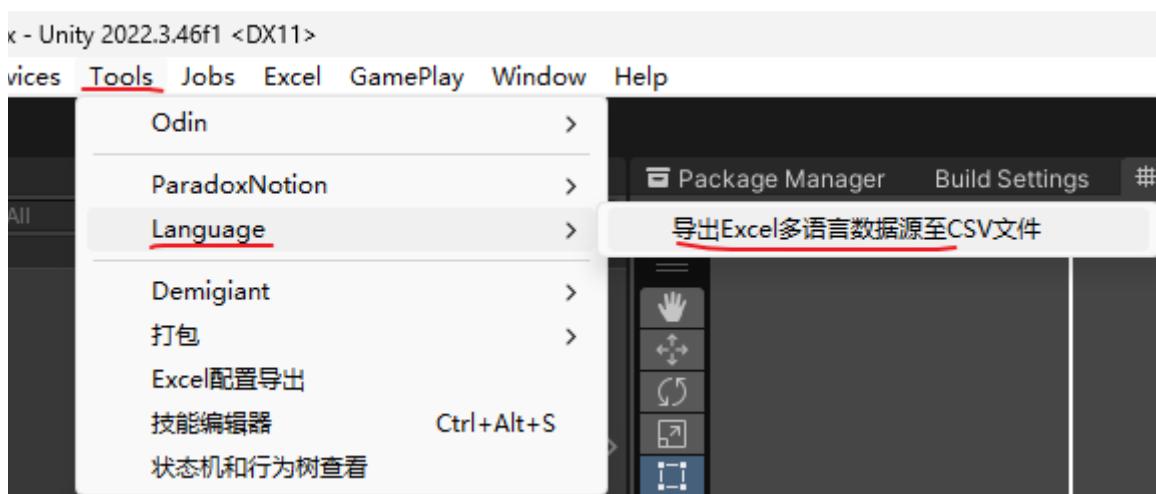
多语言的原始内容存在 excel 表里，表位于与 Assets 平级的\Other\Config\Excel\Localization 文件夹里，能任意添加 excel 表格。其中每个 excel 表格有若干个 sheet 页组成，每个 sheet 页的名字为对应的语言的名字，一般第一个 sheet 页为 Chinese 简中，目前支持的语言种类为 Chinese 简中，English 英语，Japanese 日语，Korean 汉语，TChinese 繁中。其中第一个 sheet 页 Chinese 的行列内容和其他有不同，需要特别注意。第一个 sheet 页 Chinese 中，A 列第一行为 key，B 列第一行为 Chinese，都是标题头。而别的 sheet 页，A 列第一行为 key，B 列第一行为 Chinese，C 列的第一行为该 sheet 对应的语言，比如 English 页就是 English。每列的其余行则填对应的内容，其中 key 是在 unity 里使用时需要填写的内容，然后非 Chinese 的 sheet 页的 A 列和 B 列内容应该都是从 Chinese 的 Sheet 页里复制同步而来。另外在内容列（B 列，C 列）里支持{}里填占位符 0, 1, 2, 3。。。从 0 开始，也支持##里填另一个 key 来实现名词统一。

图 1

	A	B	C	D
1	key	Chinese	English	
2	test	测试	TEST	
3	coin	金币	Coins	
4	system_get_coin	获得 {0} #coin#	Obtained {0} #coin#	
5	setting_title	设置	Setting	
6	setting_display_mode	显示模式	Display Mode	
7	setting_full_screen	全屏	Full Screen Mode	
8	setting_window	窗口	Window Mode	
9	setting_on	开启		
10	yes	是		
11	no	否		
12				

如图是一个 English 的 Sheet 也，其中 key 为 system\_get\_coin 的 key 在实际游戏 runtime 里会把{0}里的 0 替换为字符传递时的第一个参数，也会把#coin#替换为 key 对应为 coin 的本地化内容。当任意 excel 表格发生变动后，需要在 Unity 编辑器里执行导出后才能在引擎里生效。

图 2

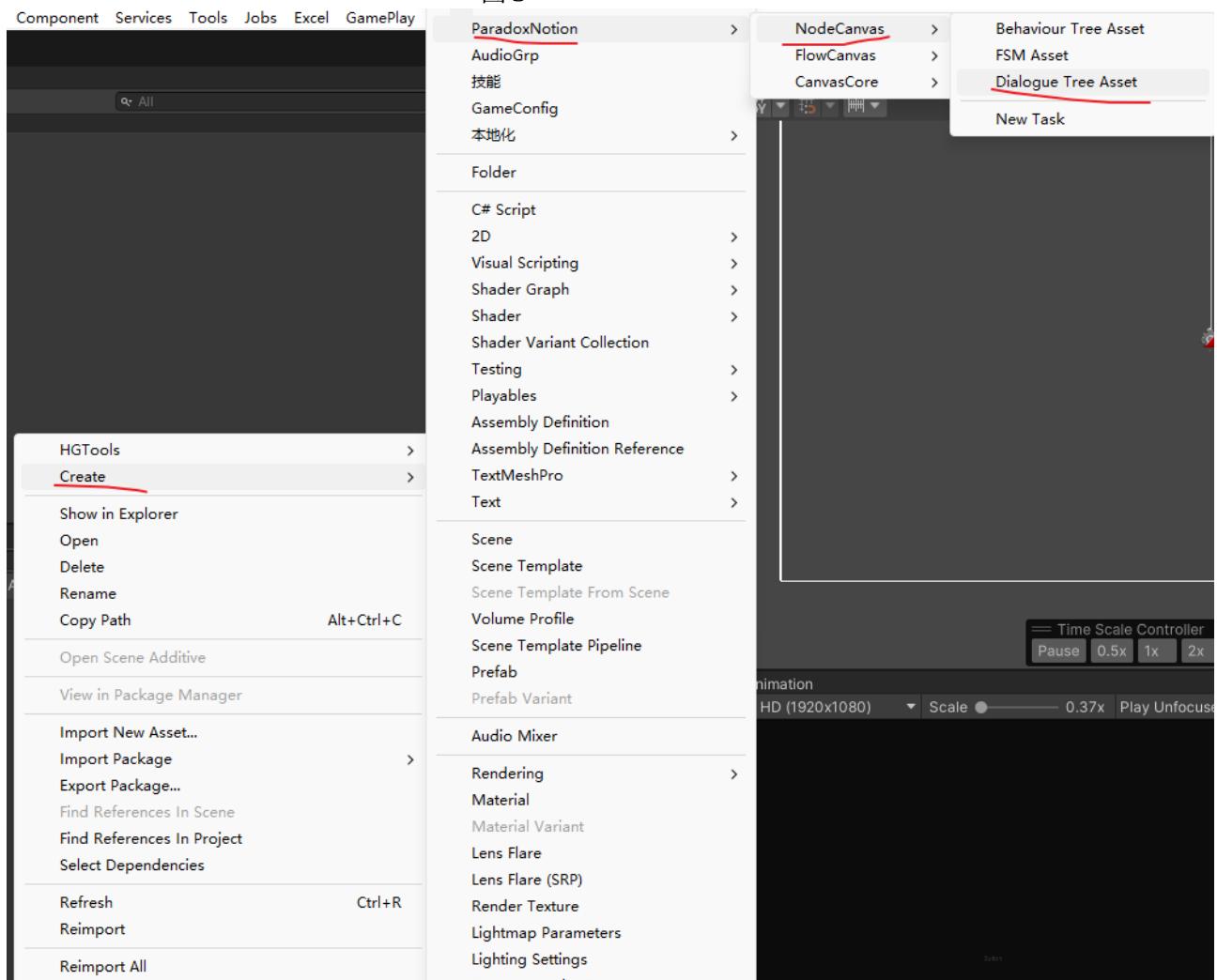


导出的文件会在\Assets\Res\ScriptableObjects\LocalizationText\Generated 里，并且会在\Assets\Res\ScriptableObjects\LocalizationSetting\DefaultLocalizationSetting.asset 里被重新引用。

## 对话

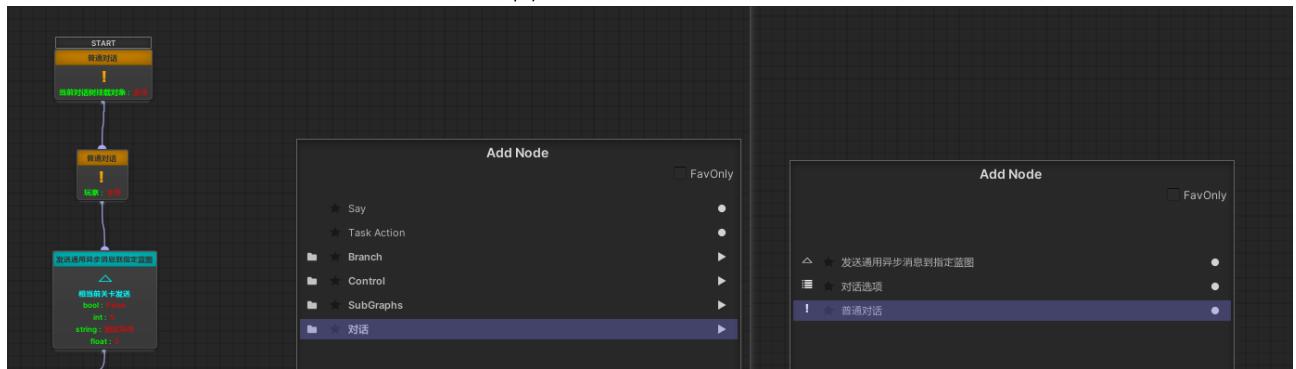
所有对话的对话树应该存放在\Assets\Res\ScriptableObjects\Dialogue 文件夹里，可以通过在Unity编辑器的Project窗口里右键新建。

图3



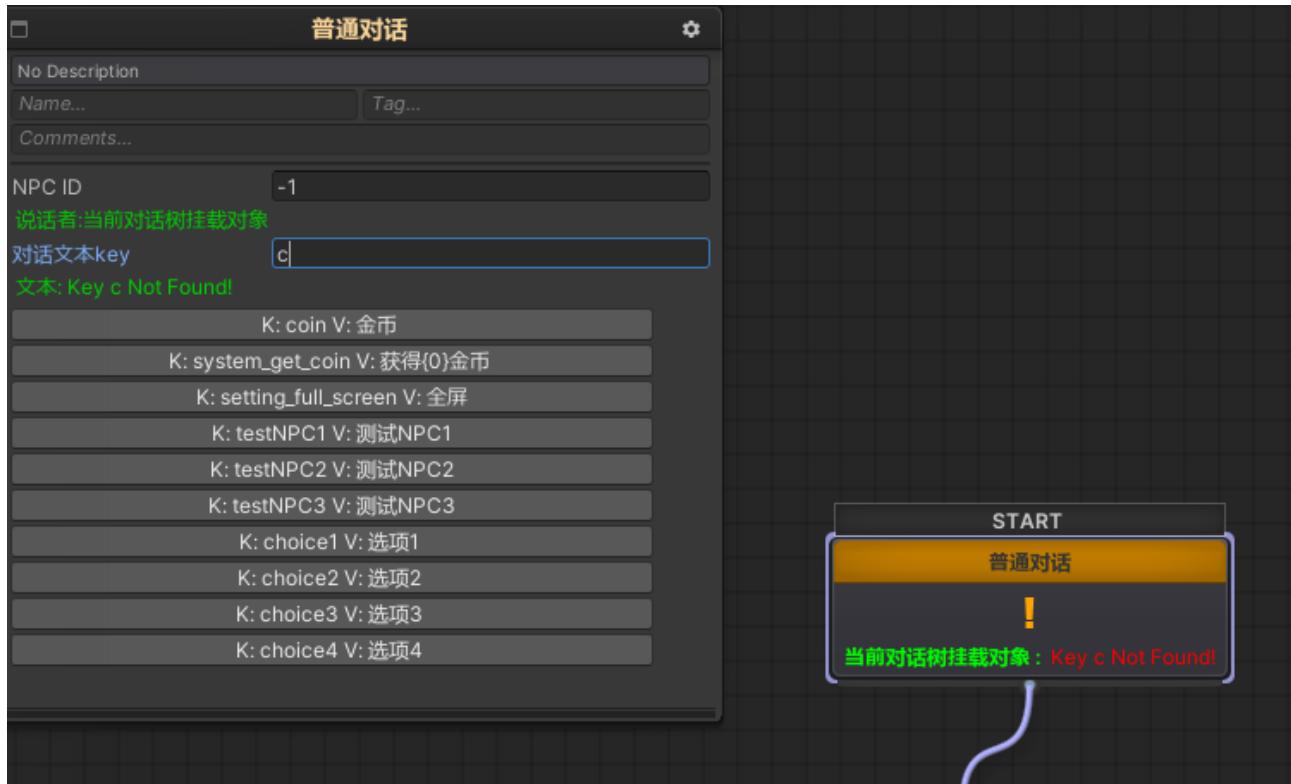
除了NodeCanvas 原生的节点之外，主要用的是项目定制的一些节点，目前提供了普通对话/选项对话/发送通用异步消息到指定蓝图 3 种定制化的节点，创建方式在打开一个对话树资源后右键-》对话里。

图4



## 普通对话节点

图 5



点开一个普通对话节点，主要填两个内容，NPC ID 和对话的本地化 key。其中 NPC ID 就是之前在\Other\Config\Excel\NPC.xlsx 里填写的 ID，也就是 gamePlay 流程里的 TableId (TId)，如果 NPC ID 填入一个负数，则认为主句化是这棵树挂载所在的 GamePlayEntity 来说，这是一个 runtime 的值。如果 NPC ID 填 0，则认为这句话是玩家来说。如果 NPC ID 填一个大于 0 的值，那会从 NPC 对应的 excel 表里找到对应的说话人。之后是 key 的填写，key 主要填写之前在\Other\Config\Excel\Localization 各个本地化 excel 里的 key 列的内容，并带有联想功能来方便选择。

## 选项对话节点

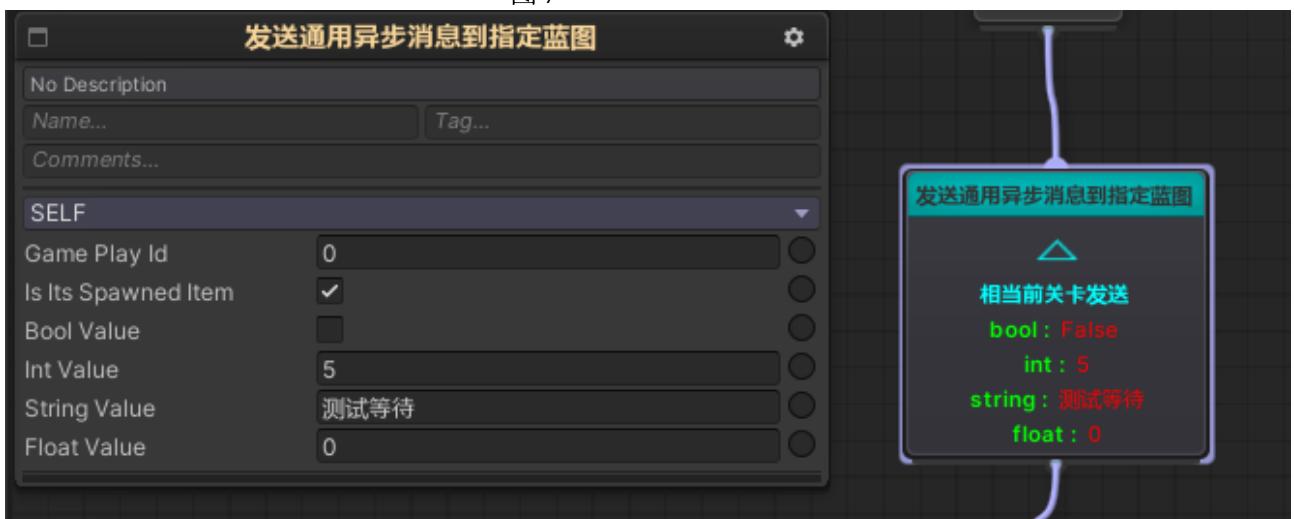
图 6



选项对话打开后和普通对话节点类似，但多了若干选项，能通过“添加选项”来新增一个选项，也能通过“X”来删除这个选项。之后可以考虑添加对话选项出现条件/倒计时时间等来增加新功能。

发送通用异步消息到指定蓝图节点

图 7



发送支持 3 个模式 Game Play Id 就是关卡里的 GamePlayId(GPID)。

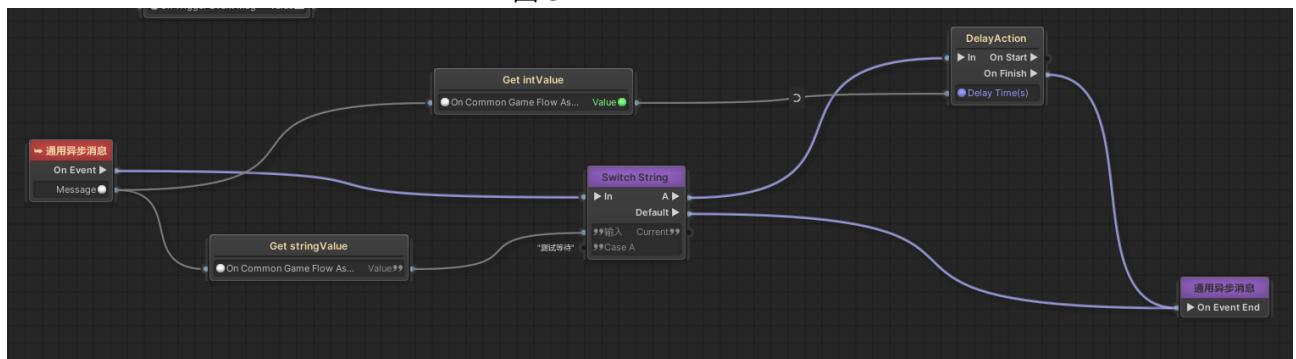
如果该值为 0，则表示这条消息发到当前关卡的关卡蓝图。

如果该值大于 0（比如填了 1000101003），并且勾选了 Is Its Spawned Item，则表示这条消息发送到当前关卡内 GPID 为 1000101003 的 GamePlayItemSpawnPoint（item 物品出生点）对应产生的 item 上，按照之前关卡的代码规则，这个 item 出生后的 GPID 应该是 1000151003。

如果该值大于 0（比如填了 1000151004），并且不勾选了 Is Its Spawned Item，则表示这条消息发送到前关卡内 GPID 为 1000151004 的 GamePlayItem。

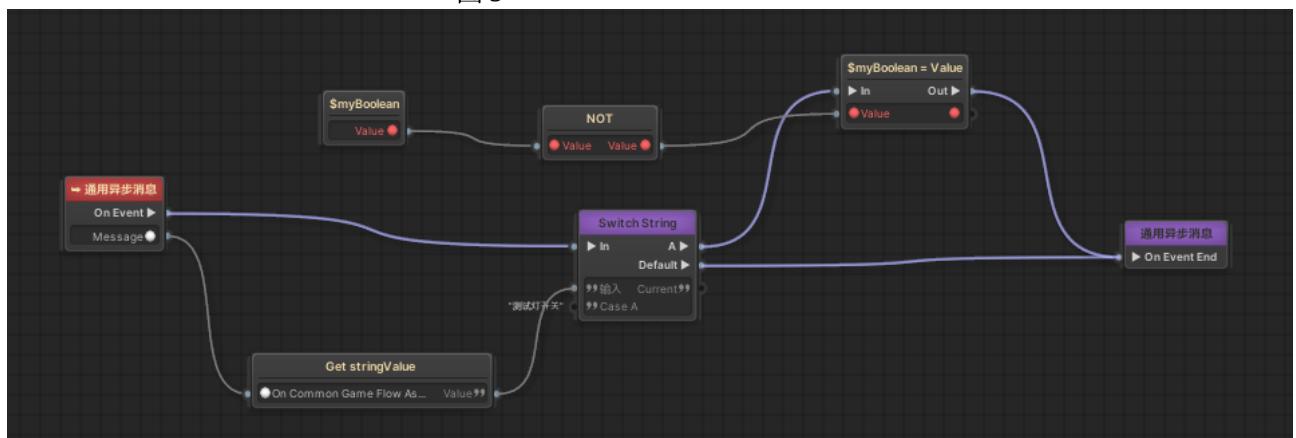
然后消息的内容提供了一个 int，一个 bool，一个 string，一个 float 的参数，能任意组合，然后在对应的消息接受的蓝图里去对应解析这条消息。注意所有的这类消息都是异步非瞬时消息，需要开始和结束对应出现，并且会在这条消息结束后再继续驱动当前对话树。

图 8



上图为如果消息里传的字符串为“测试等待”，则等待消息里的 int 值秒后结束。（该消息接收在 GamePlayTemp 这个测试用关卡的 Prefab 的关卡蓝图内）

图 9



上图为如果消息传的字符串为“测试灯开关”，则执行一次灯开关的操作（该消息接收在 TestLight 这个测试用 2D 灯光的 Prefab 的物品蓝图内）